

February TATER Monthly Status Report 2-27-2018

Contents

1 Problem Statement	1
2 Software Summary	2
2.1 Accomplishments	2
2.2 Results	2
2.3 Challenges and Next Steps	3
3 Antenna Summary	4
3.1 Pulse Larsen Hardware Design	4
3.2 Moving Forward	4
4 Hardware Summary	5
4.1 Next Steps	6
5 Key Decisions	6
6 Risks	6
6.1 Risk 0	6
6.2 Risk 1	6

1 Problem Statement

Our goal is to design a method to monitor and characterize the electromagnetic emissions of a microprocessor during boot to determine if foreign code has been injected. Limited by the types of instructions that have the most impact, the data acquisition system and analysis algorithm will be modeled accordingly. The finished product will consist of a system which can physically capture electromagnetic emissions with a custom antenna, collected using an amplifier paired with an ADC acquisition platform, and process collected data using a custom algorithm to create an EM profile of a processor's boot.

2 Software Summary

2.1 Accomplishments

The main goal was to create a program to create a model file to use as a baseline to compare against a new capture with potentially modified code. This was accomplished using a sliding window incrementing by a user-specifiable step amount over each section of the processor's boot, shown below in Figure 1.

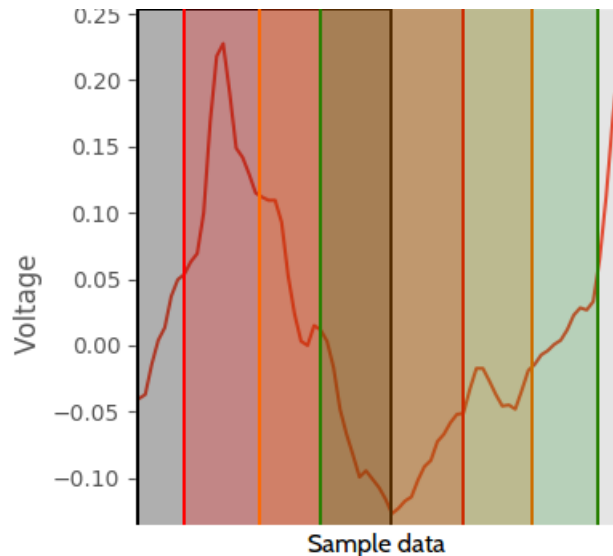


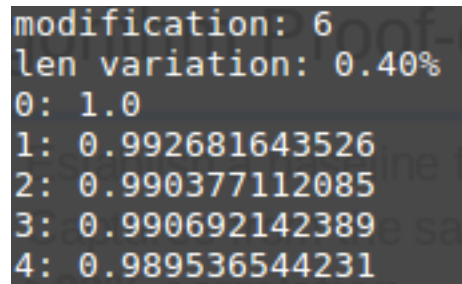
Figure 1: Window divisions separated by color. Windows overlap each other so that no region is over biased.

Rather than directly comparing the baseline to an alternate capture, windowing is used in order to ensure a brief amplitude spike caused by external noise or other factors does not influence the result. In addition, the signal is first scaled to values between -1 and 1 so that no point in the signal has a higher precedence when compared to other captures.

2.2 Results

Using the above methods, we can successfully detect modifications to the bootcode and the specific place where those modifications occur. The next goal is to characterize just how much modification can be detected, which will entail creating a list of all potential items to test.

Shown below in Figure 4 are runs of the algorithm on hardware captures taken, and how closely each sample matches to the first one. This entails a 98%-99% match between runs of the same code sample before modification.



```
modification: 6
len variation: 0.40%
0: 1.0
1: 0.992681643526
2: 0.990377112085
3: 0.990692142389
4: 0.989536544231
```

Figure 2: Same code hardware capture correlations using aforementioned method.

2.3 Challenges and Next Steps

- Changes between similar instructions (such as an ADD or MUL) are difficult to detect. For instance, the replacement of an ADD with a SUB results in close correlation; experimenting with the windowing size may be a possible mitigation.
- Adding configuration changes creates storage problems, as it is necessary to store all potential configurations and not just the single configuration change. This could entail up to multiple gigabytes depending on the length of each capture and the number of configurations.
- Some peripheral initialization times may vary significantly. This creates "dead space" within the capture where detection of foreign modification might not be possible. We hope to provide some potential workarounds and ideas for this issue in the future.

The core decision program takes a base model file and a capture and determines whether or not the capture has been modified, returning a `pass` if they are the same and a `fail` if the correlation value is less than the user specified value.

Outline of pass/fail:

- Basic input:
- Base model file: baseline created by baseline program; and capture(s): data being analyzed
 - Align both sequences (baseline and capture) to the start of the bootcode.
 - Use sliding windows and correlation on the capture data against the baseline data to determine where the modifications occur.
- Basic output: Pass or fail in console output

3 Antenna Summary

Antenna progress has surmounted to attempts at emulating the Pulse Larsen Outdoor Multi Band, and several designs drawn up in Eagle that could potentially fit the bill. The antennas drawn in Eagle need to be tested in FEKO before being fabricated, to assure quality.

3.1 Pulse Larsen Hardware Design

This hardware design is intricate, and has proven difficult to emulate, as well as too large in volume to be implemented without altering the preexisting hardware design surrounding the board.

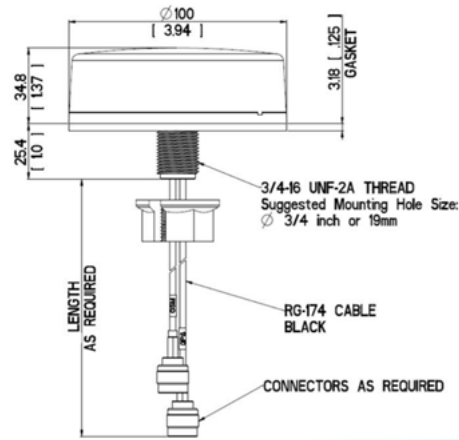


Figure 3: Hardware design for Pulse Larsen (https://www.mouser.com/catalog/specsheets/pulse_W4165XXX.pdf)

The below hardware design is modeled after the provided probes that have been being used to capture data. This is likely the design that will be ultimately implemented, with a filter to help minimize noise if need be.

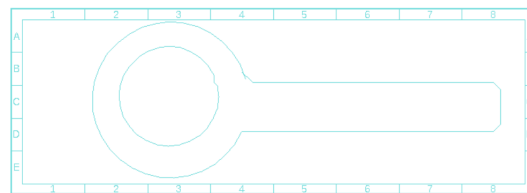


Figure 4: Custom hardware design based on probe performance.

3.2 Moving Forward

Eagle design will be modeled in FEKO, and if found to have acceptable parameters, will be crafted and implemented into the hardware design. The Pulse Larsen design will be minimized in terms of volume, as this could allow for an implementation that would not require significant hardware design alterations.

4 Hardware Summary

With the receipt of all the hardware equipment, implementation of full bootcode data capturing will now be possible. The initial setup is shown below, to be refined during progression.

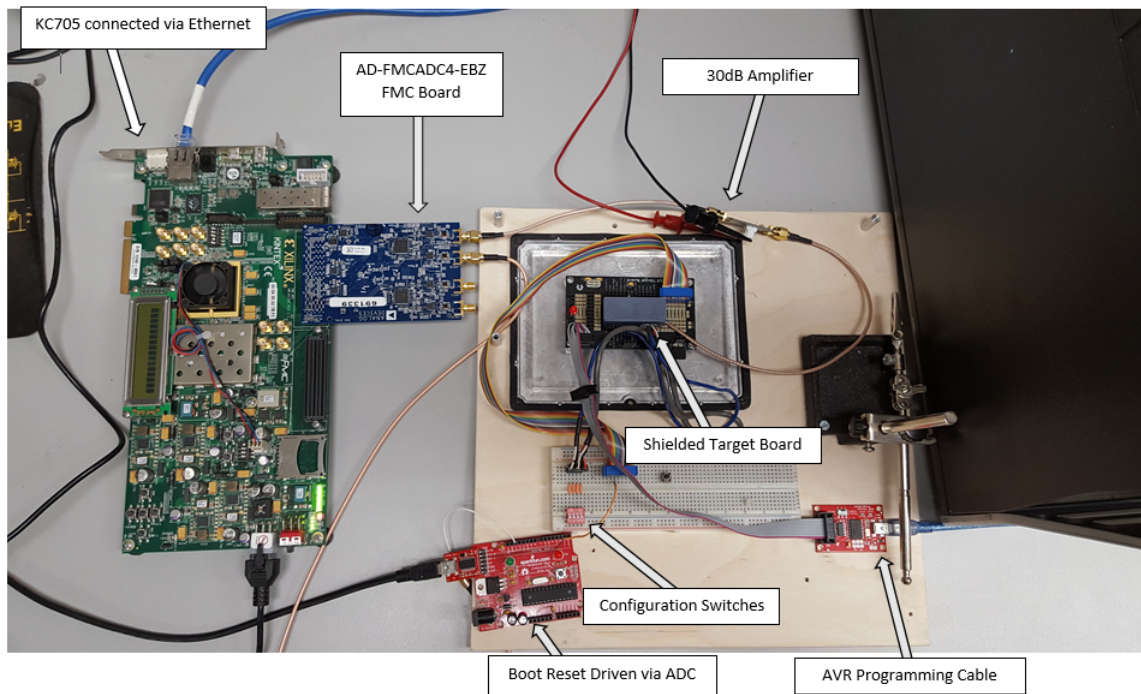


Figure 5: Platform view.

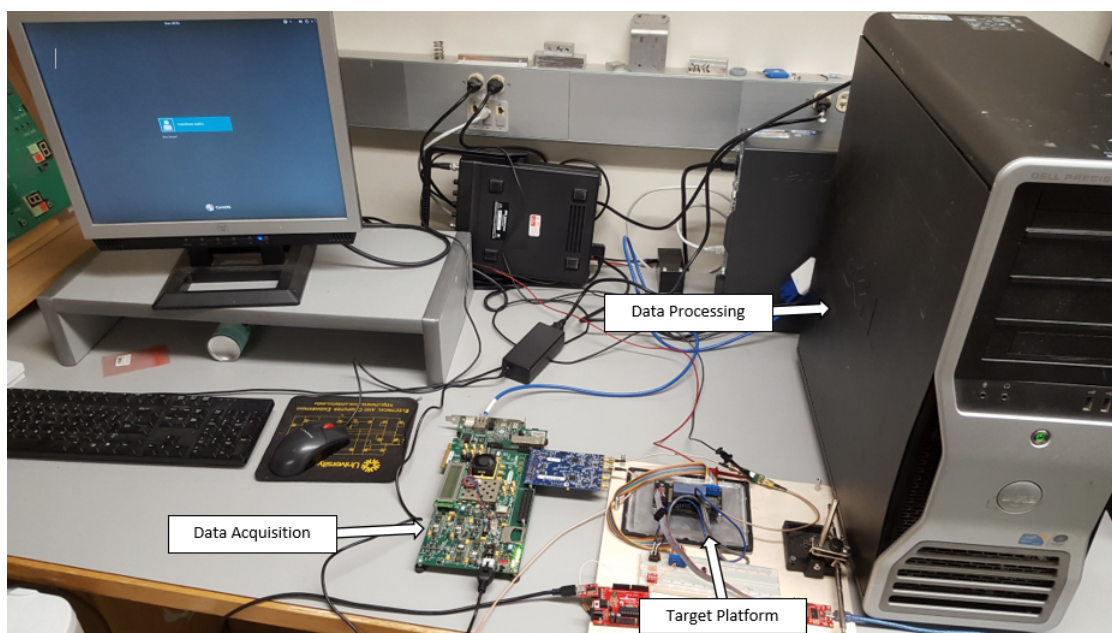


Figure 6: Setup view.

4.1 Next Steps

The next stage of implementation will be re-implementing the reference design to be compatible with the KC705, and then utilizing the Ethernet interface for data packet streaming. This will prevent us from having to create a low-level driver to interface with, and instead use the common Ethernet protocol. We will also be implementing a trigger interface to handle the reset signal from the external microcontroller.

5 Key Decisions

The team decided not to utilize preprocessing on the captures because using all the acquired captures results in a better correlation.

6 Risks

Main risks, severity, impact, and how we are mitigating the risks:

6.1 Risk 0

Antenna specifications are highly dependent on processor speed and type.

Impact: This might change the antenna size and efficiency as the center frequency and the bandwidth of the signal will be modified.

Severity: Low

Status: Building and simulating multiple antennas using our own copper will mitigate this.

6.2 Risk 1

Detection of modifications

Impact: Once we have the hardware setup implemented, we need to begin characterizing exactly what can be detected and what cannot.

Severity: High

Status: This information will also be provided during the acceptance test plan and the user guide.